Finishing up Class 10 and Class 12

Isabella Ruud: PID A59016138

Let's finish off class 10 with a pipeline for comparative structure analysis.

```
library(bio3d)
```

```
id <- "1ake_A"
aa <- get.seq(id)
```

```
Warning in get.seq(id): Removing existing file: seqs.fasta
```

```
Fetching... Please wait. Done.
```

```
aa
```

```
             1        .         .         .         .         .         60
pdb|1AKE|A   MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
             1        .         .         .         .         .         60

            61        .         .         .         .         .         120
pdb|1AKE|A   DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
            61        .         .         .         .         .         120

           121        .         .         .         .         .         180
pdb|1AKE|A   VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
           121        .         .         .         .         .         180

           181        .         .         .   214
pdb|1AKE|A   YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG
           181        .         .         .   214

Call:
```

```
  read.fasta(file = outfile)

Class:
  fasta

Alignment dimensions:
  1 sequence rows; 214 position columns (214 non-gap, 0 gap)

+ attr: id, ali, call
```

lets search for related sequences in the PDB database

```
blast <- blast.pdb(aa)
```

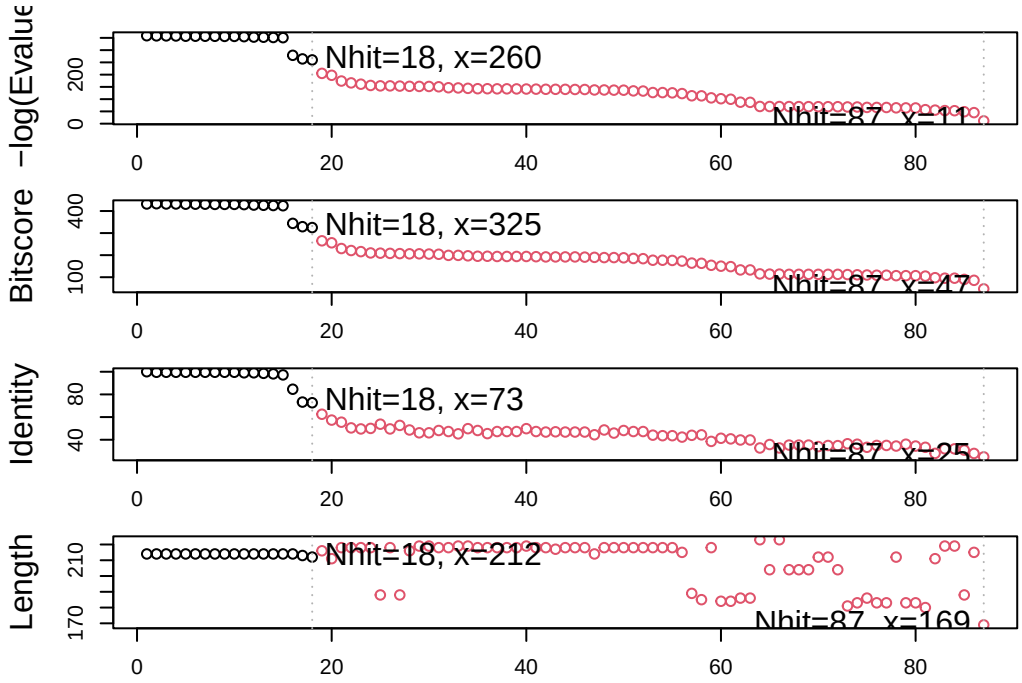
```
 Searching ... please wait (updates every 5 seconds) RID = UZT86EXX013
 .
 Reporting 87 hits
```

Let's plot an overview of the search results

```
hits <- plot(blast)
```

```
  * Possible cutoff values:    260 11
            Yielding Nhits:    18 87

  * Chosen cutoff value of:    260
            Yielding Nhits:    18
```



let's see what the top hits are

```
hits$pdb.id
```

```
 [1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "8Q2B_A" "8RJ9_A" "6RZE_A" "4X8H_A"
 [9] "3HPR_A" "1E4V_A" "5EJE_A" "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "8PVW_A"
[17] "4K46_A" "4NP6_A"
```

We can download all of these with the 'get.pdb()' function:

```
files <- get.pdb(hits$pdb.id, path="pdbs", split=TRUE, gzip=TRUE)
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1AKE.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8BQF.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8M.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6S36.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8Q2B.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8RJ9.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6RZE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8H.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3HPR.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4V.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/5EJE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4Y.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3X2S.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAP.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAM.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8PVW.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4K46.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4NP6.pdb exists. Skipping download


  |
  |                                                                      |   0%
  |
  |====                                                                  |   6%
  |
  |========                                                              |  11%
  |
  |============                                                          |  17%
  |
  |================                                                      |  22%
  |
  |====================                                                  |  28%
  |
  |========================                                              |  33%
  |
  |============================                                          |  39%
  |
  |================================                                      |  44%
  |
  |====================================                                  |  50%
  |
  |========================================                              |  56%
  |
  |============================================                          |  61%
  |
  |================================================                      |  67%
  |
  |====================================================                  |  72%
  |
  |========================================================              |  78%
  |
  |============================================================          |  83%
  |
  |================================================================      |  89%
  |
```

```
  |=================================================================     |  94%
  |
  |======================================================================| 100%
```

Align the PDBs

```
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

```
Reading PDB files:
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/8BQF_A.pdb
pdbs/split_chain/4X8M_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/8Q2B_A.pdb
pdbs/split_chain/8RJ9_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/4X8H_A.pdb
pdbs/split_chain/3HPR_A.pdb
pdbs/split_chain/1E4V_A.pdb
pdbs/split_chain/5EJE_A.pdb
pdbs/split_chain/1E4Y_A.pdb
pdbs/split_chain/3X2S_A.pdb
pdbs/split_chain/6HAP_A.pdb
pdbs/split_chain/6HAM_A.pdb
pdbs/split_chain/8PVW_A.pdb
pdbs/split_chain/4K46_A.pdb
pdbs/split_chain/4NP6_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..   PDB has ALT records, taking A only, rm.alt=TRUE
..   PDB has ALT records, taking A only, rm.alt=TRUE
....   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..

Extracting sequences
```

```
pdb/seq: 1   name: pdbs/split_chain/1AKE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 2   name: pdbs/split_chain/8BQF_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 3   name: pdbs/split_chain/4X8M_A.pdb
pdb/seq: 4   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 5   name: pdbs/split_chain/8Q2B_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 6   name: pdbs/split_chain/8RJ9_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 7   name: pdbs/split_chain/6RZE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 8   name: pdbs/split_chain/4X8H_A.pdb
pdb/seq: 9   name: pdbs/split_chain/3HPR_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 10   name: pdbs/split_chain/1E4V_A.pdb
pdb/seq: 11   name: pdbs/split_chain/5EJE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 12   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 13   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 14   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 15   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 16   name: pdbs/split_chain/8PVW_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 17   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 18   name: pdbs/split_chain/4NP6_A.pdb
```

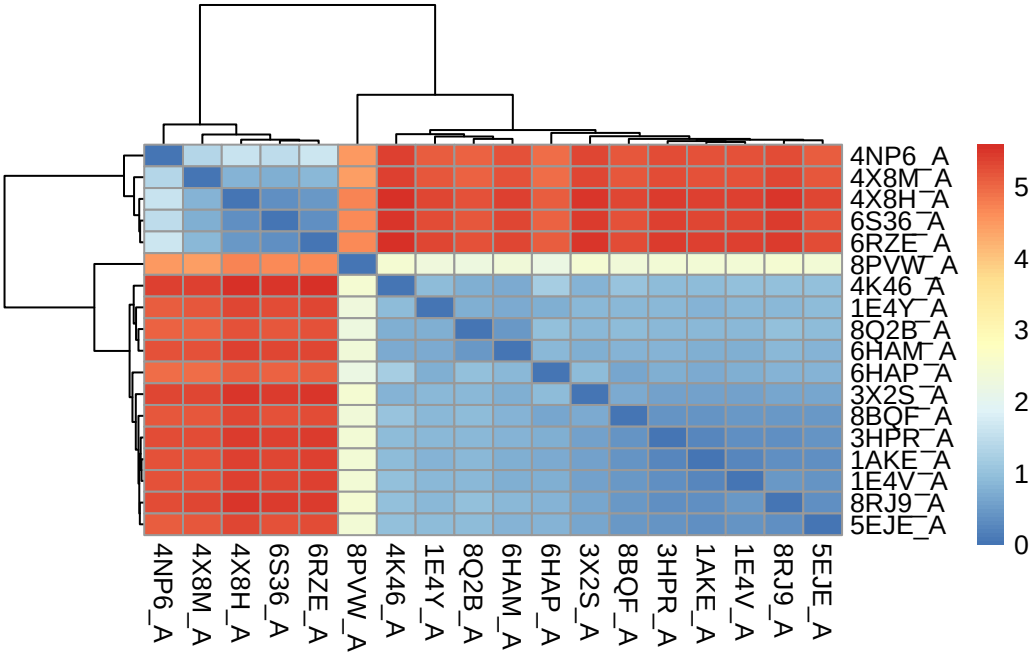
Conventional analysis

An RMSD analysis

```
rd <- rmsd(pdbs)
```

```
Warning in rmsd(pdbs): No indices provided, using the 182 non NA positions
```

```
library(pheatmap)
pheatmap(rd)
```




```
source("https://tinyurl.com/newviewngl")
library(NGLVieweR)
```

```
#view.pdbs(pdbs)
```

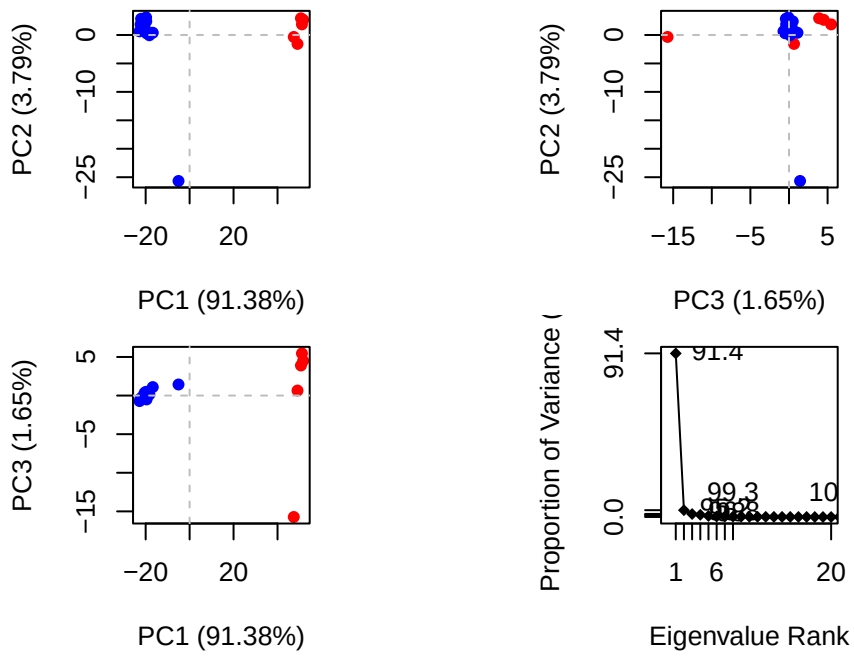
We can cluster by RMSD and then color our structure display

```
km <- kmeans(rd, centers=2)
mycols <- c("red", "blue")[km$cluster]
```

```
#view.pdbs(pdbs, color = mycols)
```

Principal Component analysis

```
pc <- pca(pdbs)
plot(pc, col=mycols)
```




```
p <- mktrj(pc, file = "pca.pdb")
p <- read.pdb("pca.pdb", multi = T)
#view.pdb(p)
```

```
ref <- read.pdb("1ake")
```

```
  Note: Accessing on-line PDB file
   PDB has ALT records, taking A only, rm.alt=TRUE
```

```
#view.pdb(ref)
```

AlphaFold Analysis

Here we demonstrate how to analyze and make sense of models from AlphaFold. We begin by reading in all the model PDB files

```
library(bio3d)
files <- list.files("./hiv_monomer_94b5b", pattern = ".pdb", full.names = T)
files
```

```
[1] "./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_001_alphafold2_ptm_model_5_seed_000
[2] "./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000
[3] "./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_003_alphafold2_ptm_model_1_seed_000
[4] "./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_004_alphafold2_ptm_model_3_seed_000
[5] "./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000
```

align and superpose

```
pbds <- pdbaln(files, fit=TRUE, exefile = "msa")
```

```
Reading PDB files:
./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_001_alphafold2_ptm_model_5_seed_000.pdb
./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000.pdb
./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_003_alphafold2_ptm_model_1_seed_000.pdb
./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_004_alphafold2_ptm_model_3_seed_000.pdb
./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb
.....

Extracting sequences

pdb/seq: 1   name: ./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_001_alphafold2_ptm_m
pdb/seq: 2   name: ./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_002_alphafold2_ptm_m
pdb/seq: 3   name: ./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_003_alphafold2_ptm_m
pdb/seq: 4   name: ./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_004_alphafold2_ptm_m
pdb/seq: 5   name: ./hiv_monomer_94b5b/hiv_monomer_94b5b_unrelaxed_rank_005_alphafold2_ptm_m
```

```
source("https://tinyurl.com/newviewngl")
library(NGLVieweR)
#view.pdbs(pdbs)
```

RMSD analysis

```
rd <- rmsd(pdbs)
```

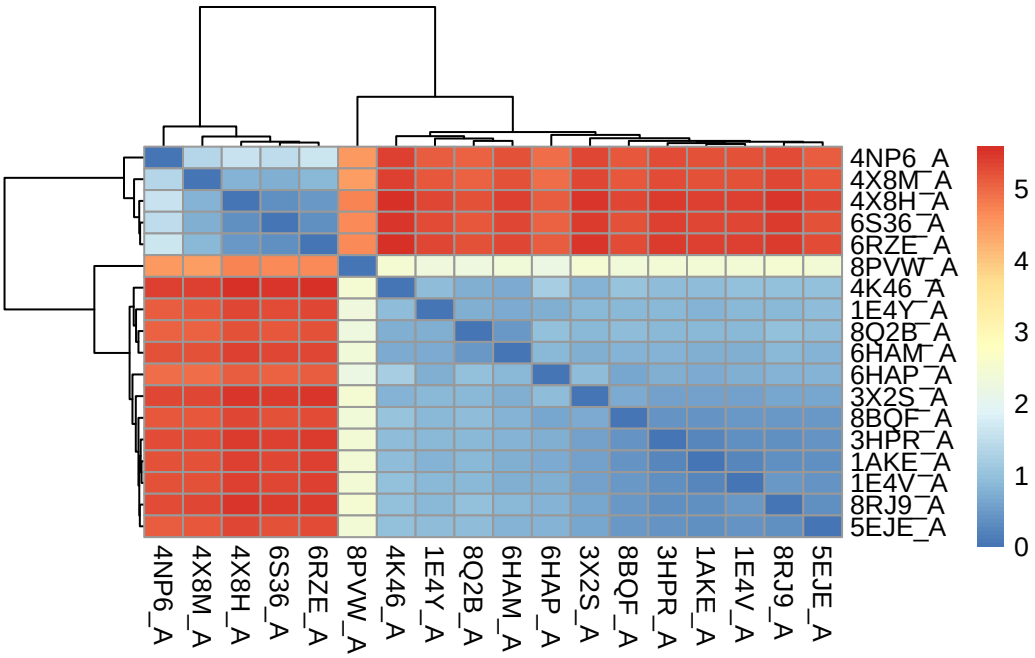
```
Warning in rmsd(pdbs): No indices provided, using the 182 non NA positions
```

```
summary(rd)
```

```
     1AKE_A           8BQF_A           4X8M_A          6S36_A
 Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.000
 1st Qu.:0.3795   1st Qu.:0.4765   1st Qu.:2.129   1st Qu.:2.292
 Median :0.7900   Median :0.8510   Median :5.146   Median :5.245
 Mean   :1.9610   Mean   :1.9888   Mean   :3.925   Mean   :3.961
 3rd Qu.:4.5050   3rd Qu.:4.4467   3rd Qu.:5.231   3rd Qu.:5.322
 Max.   :5.4020   Max.   :5.3220   Max.   :5.394   Max.   :5.505
     8Q2B_A           8RJ9_A           6RZE_A          4X8H_A
 Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.000
 1st Qu.:0.8565   1st Qu.:0.4592   1st Qu.:2.382   1st Qu.:2.357
 Median :0.9135   Median :0.8500   Median :5.293   Median :5.326
 Mean   :2.0693   Mean   :2.0339   Mean   :4.010   Mean   :4.030
 3rd Qu.:4.3525   3rd Qu.:4.5898   3rd Qu.:5.368   3rd Qu.:5.401
 Max.   :5.2460   Max.   :5.4960   Max.   :5.547   Max.   :5.583
     3HPR_A           1E4V_A           5EJE_A           1E4Y_A
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.4078   1st Qu.:0.4547   1st Qu.:0.4340   1st Qu.:0.8297
 Median :0.8145   Median :0.8165   Median :0.8715   Median :0.8760
 Mean   :1.9912   Mean   :1.9833   Mean   :1.9852   Mean   :2.0953
 3rd Qu.:4.5445   3rd Qu.:4.5062   3rd Qu.:4.4553   3rd Qu.:4.4322
 Max.   :5.4510   Max.   :5.3980   Max.   :5.3300   Max.   :5.3450
     3X2S_A           6HAP_A           6HAM_A           8PVW_A
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
 1st Qu.:0.6565   1st Qu.:0.7685   1st Qu.:0.7568   1st Qu.:2.377
 Median :0.8280   Median :0.8930   Median :0.8290   Median :2.422
 Mean   :2.0889   Mean   :2.0318   Mean   :2.0604   Mean   :2.866
 3rd Qu.:4.5998   3rd Qu.:4.2477   3rd Qu.:4.4908   3rd Qu.:3.949
 Max.   :5.5210   Max.   :5.1230   Max.   :5.3800   Max.   :4.713
     4K46_A          4NP6_A
 Min.   :0.000   Min.   :0.000
 1st Qu.:0.920   1st Qu.:2.344
 Median :0.976   Median :5.135
 Mean   :2.226   Mean   :4.045
 3rd Qu.:4.652   3rd Qu.:5.205
 Max.   :5.583   Max.   :5.373
```

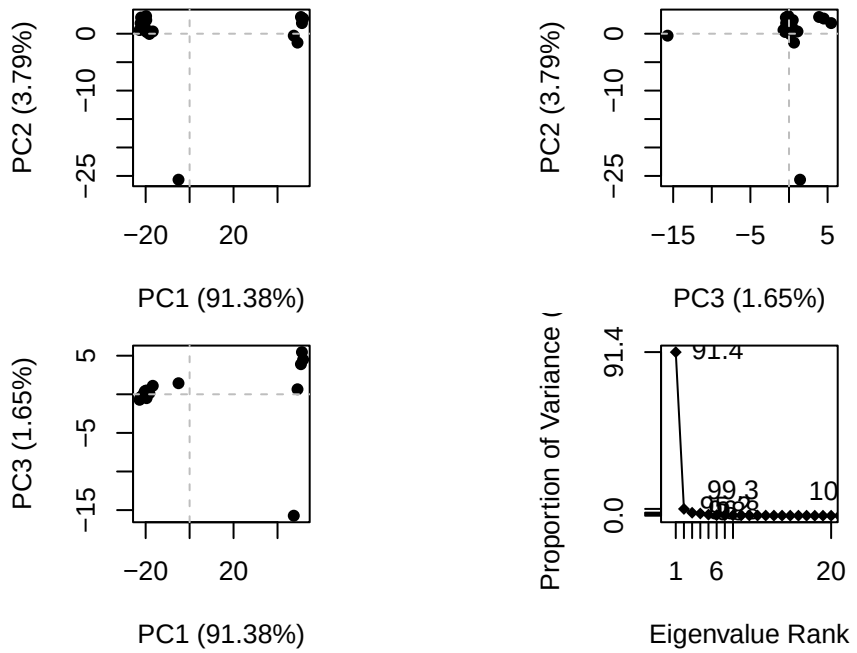
```
library(pheatmap)

pheatmap(rd)
```



PCA

```
pc <- pca(pdbs)
plot(pc)
```



Residue conservation from alignment file

AlphaFold writes out the MSA it calculated and used for structure prediction to a A3M format file that we can read into R for further analysis

```
aln_file <- list.files("hiv_monomer_94b5b/", pattern = ".a3m$", full.names = TRUE)
aln_file
```

```
[1] "hiv_monomer_94b5b/hiv_monomer_94b5b.a3m"
```

```
aln <- read.fasta(aln_file, to.upper=TRUE)
```

```
[1] " ** Duplicated sequence id's: 101 **"
```

```
dim(aln$ali)
```

```
[1] 5378  132
```

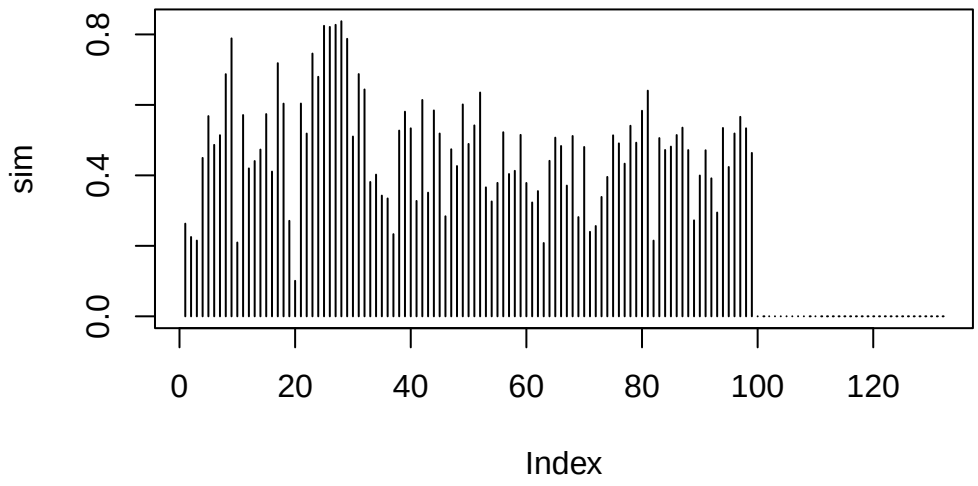
```
sim <- conserv(aln)
```

```
con <- consensus(aln, cutoff = 0.9)
con$seq
```

```
  [1] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [19] "-" "-" "-" "-" "-" "-" "D" "T" "G" "A" "-" "-" "-" "-" "-" "-" "-" "-"
 [37] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [55] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [73] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [91] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[109] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[127] "-" "-" "-" "-" "-" "-"
```

Plot the conservation along the sequence/structure

```
plot(sim,typ="h")
```



Let's look at these conserved positions

```
pdb <- read.pdb(files[1])
#view.pdb(pdb, backgroundColor="pink",
         #highlight = atom.select(pdb, resno=25:28),   highlight.style="spacefill")
```

HIV dimer

Read in the pdb files that alphafold gave

```
results_dir <- "hiv_dimer_23119.result/hiv_dimer_23119/"
pdb_files <- list.files(path=results_dir,
                        pattern="*.pdb",
                        full.names = TRUE)

# Print our PDB file names
basename(pdb_files)
```

```
[1] "hiv_dimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb"
[2] "hiv_dimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb"
[3] "hiv_dimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb"
[4] "hiv_dimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb"
```

```
[5] "hiv_dimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb"
```

superimpose the data from the models

```
library(bio3d)

# Read all data from Models
#  and superpose/fit coords
pdbs <- pdbaln(pdb_files, fit=TRUE, exefile="msa")
```

```
Reading PDB files:
hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_001_alphafold2_multimer
hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_002_alphafold2_multimer
hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_003_alphafold2_multimer
hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_004_alphafold2_multimer
hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_005_alphafold2_multimer
.....

Extracting sequences

pdb/seq: 1   name: hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_001_
pdb/seq: 2   name: hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_002_
pdb/seq: 3   name: hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_003_
pdb/seq: 4   name: hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_004_
pdb/seq: 5   name: hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119_unrelaxed_rank_005_
```

Find the RMSD between the models

```
rd <- rmsd(pdbs, fit=T)
```

```
Warning in rmsd(pdbs, fit = T): No indices provided, using the 198 non NA positions
```

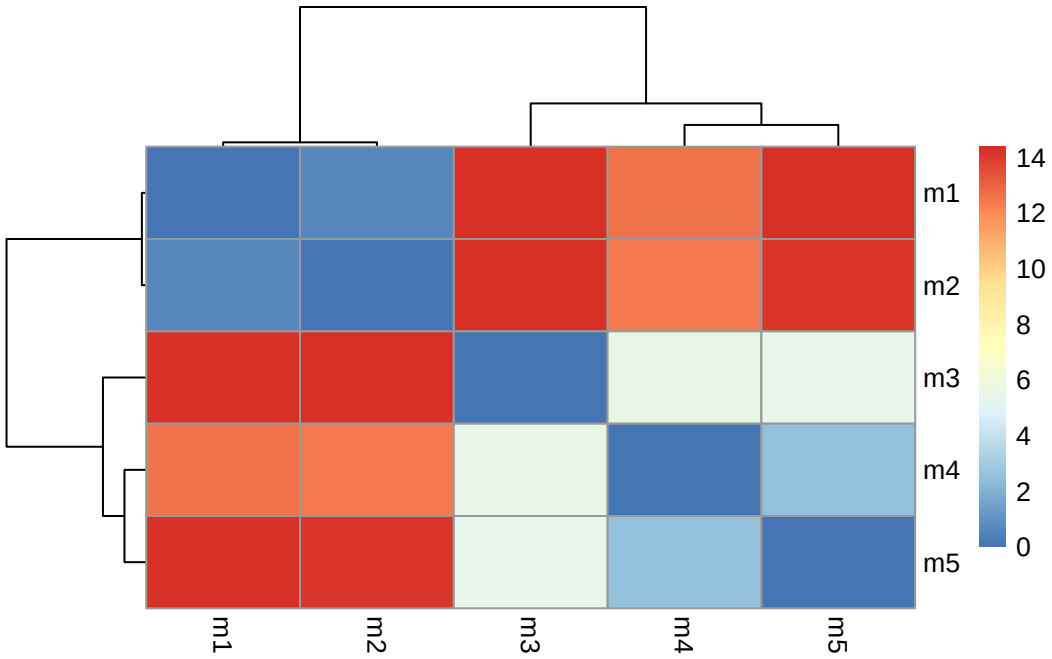
```
range(rd)
```

```
[1]  0.00 14.37
```

Visualize the RMSD between the models as a heatmap

```
library(pheatmap)

colnames(rd) <- paste0("m",1:5)
rownames(rd) <- paste0("m",1:5)
pheatmap(rd)
```



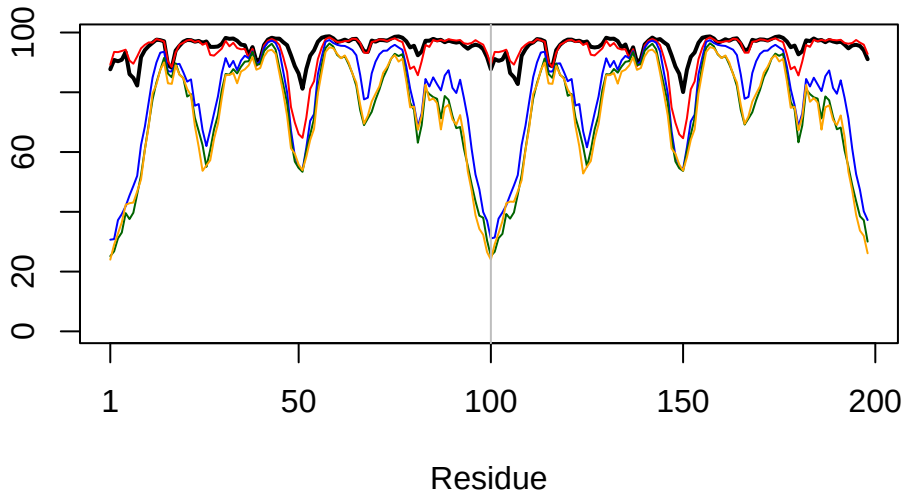
Plot the pLDDT values across the models

```
plotb3(pdbs$b[1,], typ="l", lwd=2, sse=pdb)
```

```
Warning in pdb2sse(sse): No helix and sheet defined in input 'sse' PDB object:
try using dssp()
```

```
Warning in plotb3(pdbs$b[1, ], typ = "l", lwd = 2, sse = pdb): Length of input
'sse' does not equal the length of input 'x'; Ignoring 'sse'
```

```
points(pdbs$b[2,], typ="l", col="red")
points(pdbs$b[3,], typ="l", col="blue")
points(pdbs$b[4,], typ="l", col="darkgreen")
points(pdbs$b[5,], typ="l", col="orange")
abline(v=100, col="gray")
```



Improve the superimposition by finding a rigid core

```
core <- core.find(pdbs)
```

```
 core size 197 of 198  vol = 4941.42
 core size 196 of 198  vol = 4329.949
 core size 195 of 198  vol = 4120.446
 core size 194 of 198  vol = 3924.504
 core size 193 of 198  vol = 3727.787
 core size 192 of 198  vol = 3555.457
 core size 191 of 198  vol = 3451.357
 core size 190 of 198  vol = 3345.941
 core size 189 of 198  vol = 3257.346
 core size 188 of 198  vol = 3164.397
 core size 187 of 198  vol = 3080.193
 core size 186 of 198  vol = 3036.325
 core size 185 of 198  vol = 2980.825
 core size 184 of 198  vol = 2956.26
 core size 183 of 198  vol = 2923.042
 core size 182 of 198  vol = 2887.361
 core size 181 of 198  vol = 2879.837
 core size 180 of 198  vol = 2916.544
 core size 179 of 198  vol = 2938.95
```

```
core size 178 of 198  vol = 2974.744
core size 177 of 198  vol = 3040.008
core size 176 of 198  vol = 3077.168
core size 175 of 198  vol = 3115.019
core size 174 of 198  vol = 3147.56
core size 173 of 198  vol = 3138.972
core size 172 of 198  vol = 3098.495
core size 171 of 198  vol = 3043.363
core size 170 of 198  vol = 2999.212
core size 169 of 198  vol = 2951.369
core size 168 of 198  vol = 2852.115
core size 167 of 198  vol = 2761.006
core size 166 of 198  vol = 2681.221
core size 165 of 198  vol = 2607.04
core size 164 of 198  vol = 2538.506
core size 163 of 198  vol = 2465.366
core size 162 of 198  vol = 2388.894
core size 161 of 198  vol = 2322.105
core size 160 of 198  vol = 2237.107
core size 159 of 198  vol = 2156.996
core size 158 of 198  vol = 2078.016
core size 157 of 198  vol = 2007.35
core size 156 of 198  vol = 1943.806
core size 155 of 198  vol = 1863.421
core size 154 of 198  vol = 1786.389
core size 153 of 198  vol = 1708.495
core size 152 of 198  vol = 1635.981
core size 151 of 198  vol = 1560.01
core size 150 of 198  vol = 1488.246
core size 149 of 198  vol = 1422.735
core size 148 of 198  vol = 1363.242
core size 147 of 198  vol = 1309.443
core size 146 of 198  vol = 1265.637
core size 145 of 198  vol = 1218.319
core size 144 of 198  vol = 1163.274
core size 143 of 198  vol = 1122.077
core size 142 of 198  vol = 1082.585
core size 141 of 198  vol = 1037.275
core size 140 of 198  vol = 994.581
core size 139 of 198  vol = 952.675
core size 138 of 198  vol = 904.481
core size 137 of 198  vol = 865.673
core size 136 of 198  vol = 838.094
```

```
core size 135 of 198  vol = 804.573
core size 134 of 198  vol = 773.607
core size 133 of 198  vol = 735.249
core size 132 of 198  vol = 697.903
core size 131 of 198  vol = 657.676
core size 130 of 198  vol = 621.282
core size 129 of 198  vol = 585.818
core size 128 of 198  vol = 550.423
core size 127 of 198  vol = 517.835
core size 126 of 198  vol = 489.513
core size 125 of 198  vol = 455.018
core size 124 of 198  vol = 425.108
core size 123 of 198  vol = 402.456
core size 122 of 198  vol = 390.648
core size 121 of 198  vol = 379.116
core size 120 of 198  vol = 350.091
core size 119 of 198  vol = 318.854
core size 118 of 198  vol = 293.293
core size 117 of 198  vol = 269.048
core size 116 of 198  vol = 247.877
core size 115 of 198  vol = 232.089
core size 114 of 198  vol = 210.894
core size 113 of 198  vol = 191.752
core size 112 of 198  vol = 169.053
core size 111 of 198  vol = 149.363
core size 110 of 198  vol = 135.803
core size 109 of 198  vol = 125.099
core size 108 of 198  vol = 113.647
core size 107 of 198  vol = 103.994
core size 106 of 198  vol = 96.29
core size 105 of 198  vol = 89.447
core size 104 of 198  vol = 82.868
core size 103 of 198  vol = 76.649
core size 102 of 198  vol = 70.575
core size 101 of 198  vol = 65.752
core size 100 of 198  vol = 61.956
core size 99 of 198  vol = 58.908
core size 98 of 198  vol = 55.216
core size 97 of 198  vol = 49.559
core size 96 of 198  vol = 43.279
core size 95 of 198  vol = 36.722
core size 94 of 198  vol = 31.361
core size 93 of 198  vol = 23.174
```

```
 core size 92 of 198  vol = 15.633
 core size 91 of 198  vol = 9.523
 core size 90 of 198  vol = 5.156
 core size 89 of 198  vol = 3.429
 core size 88 of 198  vol = 2.931
 core size 87 of 198  vol = 2.441
 core size 86 of 198  vol = 1.99
 core size 85 of 198  vol = 1.63
 core size 84 of 198  vol = 1.389
 core size 83 of 198  vol = 1.144
 core size 82 of 198  vol = 0.934
 core size 81 of 198  vol = 0.806
 core size 80 of 198  vol = 0.664
 core size 79 of 198  vol = 0.602
 core size 78 of 198  vol = 0.532
 core size 77 of 198  vol = 0.486
 FINISHED: Min vol ( 0.5 ) reached
```

```
core.inds <- print(core, vol=0.5)
```

```
# 78 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1    10  25     16
2    28  48     21
3    53  93     41
```

```
xyz <- pdbfit(pdbs, core.inds, outpath="corefit_structures")
```

Look at RMSF between positions on the structure

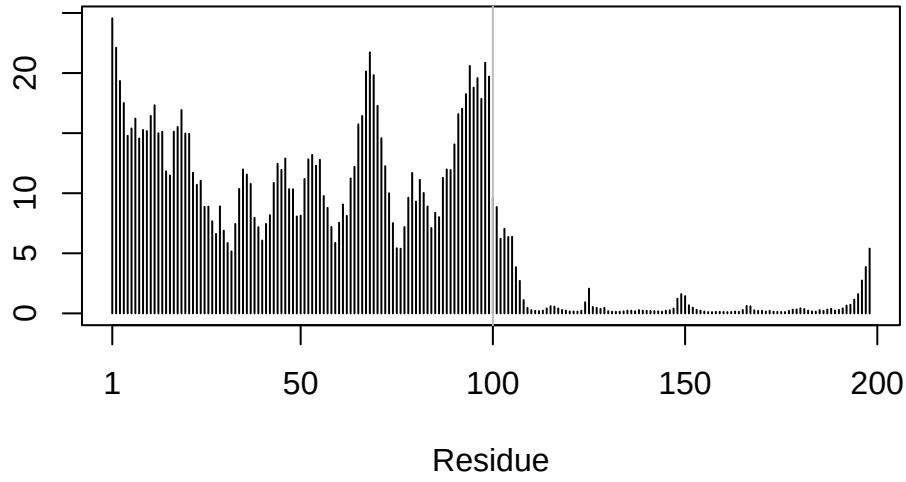
```
rf <- rmsf(xyz)

plotb3(rf, sse=pdb)
```

```
Warning in pdb2sse(sse): No helix and sheet defined in input 'sse' PDB object:
try using dssp()
```

```
Warning in plotb3(rf, sse = pdb): Length of input 'sse' does not equal the
length of input 'x'; Ignoring 'sse'
```

```
abline(v=100, col="gray", ylab="RMSF")
```



Predicted alignment error for domains

```
library(jsonlite)

# Listing of all PAE JSON files
pae_files <- list.files(path=results_dir,
                        pattern=".*model.*\\.json",
                        full.names = TRUE)
```

```
pae1 <- read_json(pae_files[1],simplifyVector = TRUE)
pae5 <- read_json(pae_files[5],simplifyVector = TRUE)

attributes(pae1)
```

```
$names
[1] "plddt"   "max_pae" "pae"     "ptm"     "iptm"
```

```
# Per-residue pLDDT scores
#  same as B-factor of PDB..
head(pae1$plddt)
```

```
[1] 87.69 90.81 90.38 90.88 93.44 86.06
```

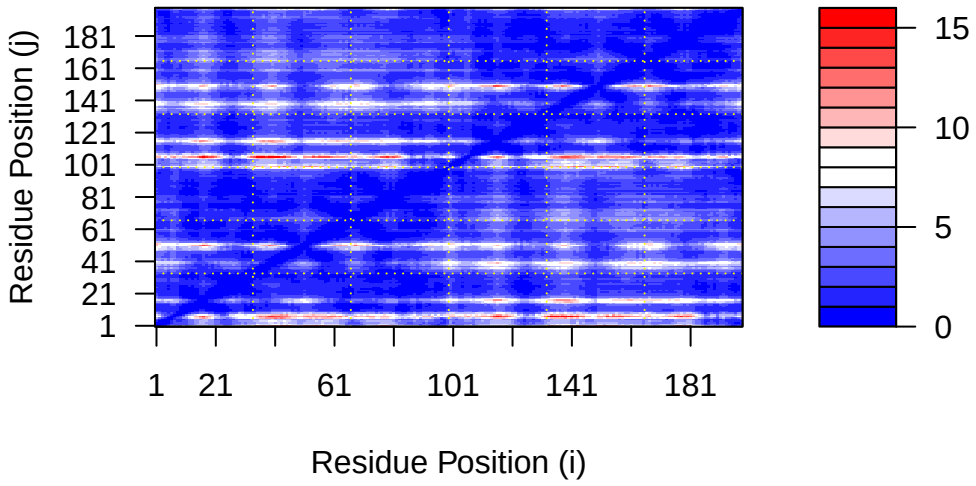
```
pae1$max_pae
```

```
[1] 15.47656
```

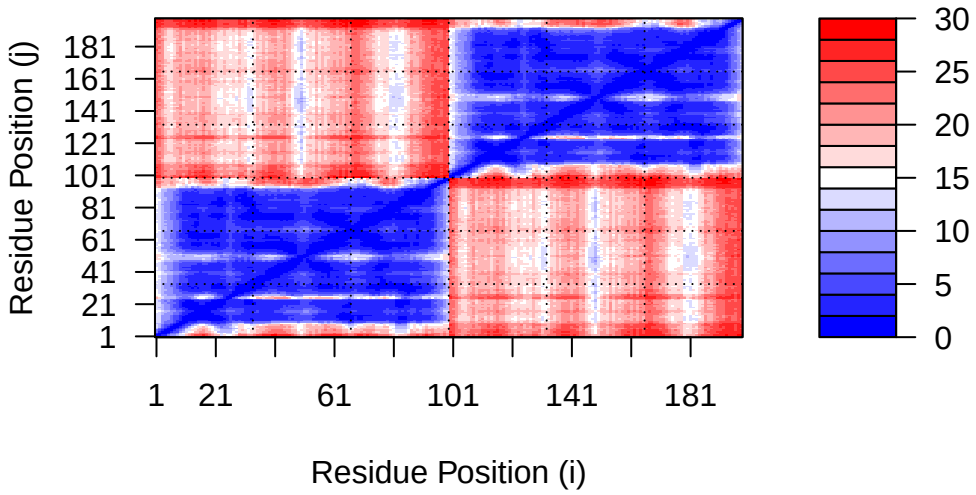
```
pae5$max_pae
```

```
[1] 29.32812
```

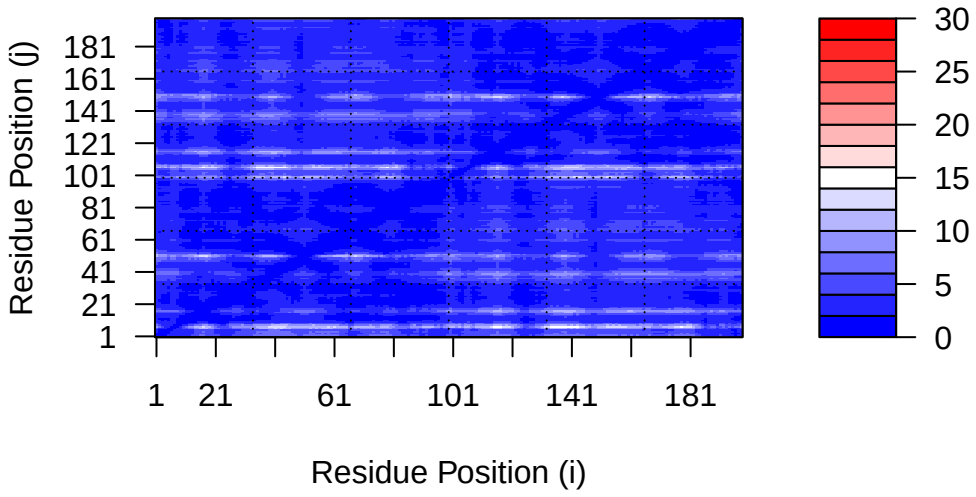
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```




```
plot.dmat(pae5$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim=c(0,30))
```




```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim=c(0,30))
```



Residue conservation from alignment file

```
aln_file <- list.files(path=results_dir,
                       pattern=".a3m$",
                        full.names = TRUE)
aln_file
```

```
[1] "hiv_dimer_23119.result/hiv_dimer_23119/hiv_dimer_23119.a3m"
```

```
aln <- read.fasta(aln_file[1], to.upper = TRUE)
```

```
[1] " ** Duplicated sequence id's: 101 **"
[2] " ** Duplicated sequence id's: 101 **"
```

```
dim(aln$ali)
```

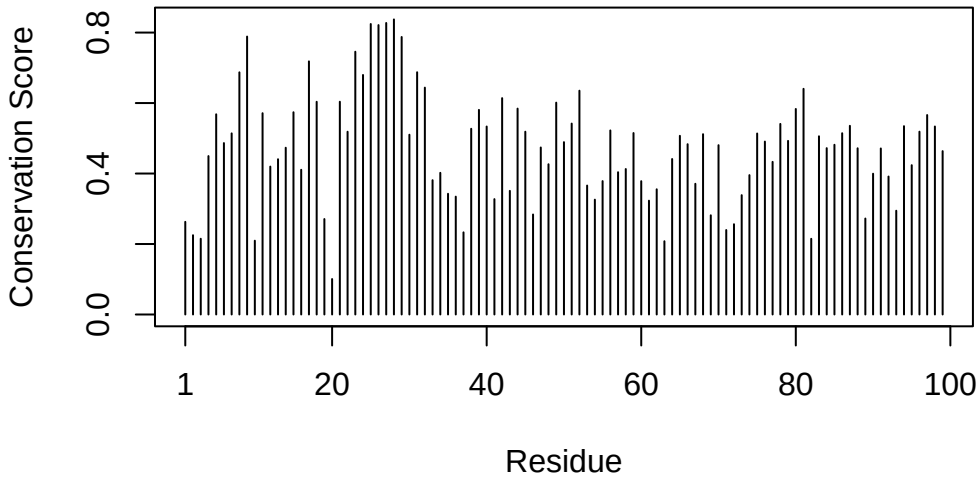
```
[1] 5378  132
```

```
sim <- conserv(aln)
```

```
plotb3(sim[1:99], sse=trim.pdb(pdb, chain="A"),
       ylab="Conservation Score")
```

```
Warning in pdb2sse(sse): No helix and sheet defined in input 'sse' PDB object:
try using dssp()
```

```
Warning in plotb3(sim[1:99], sse = trim.pdb(pdb, chain = "A"), ylab =
"Conservation Score"): Length of input 'sse' does not equal the length of input
'x'; Ignoring 'sse'
```




```
con <- consensus(aln, cutoff = 0.9)
con$seq
```

```
  [1] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [19] "-" "-" "-" "-" "-" "-" "D" "T" "G" "A" "-" "-" "-" "-" "-" "-" "-" "-"
 [37] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [55] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [73] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [91] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[109] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[127] "-" "-" "-" "-" "-" "-"
```

```
m1.pdb <- read.pdb(pdb_files[1])
occ <- vec2resno(c(sim[1:99], sim[1:99]), m1.pdb$atom$resno)
write.pdb(m1.pdb, o=occ, file="m1_conserv.pdb")
```